A Web Service-Based Approach to Integrated Information Management for Railway Emergency Scenario

ZHANG Zhen-Hai[1*], XU Zhi-Long[1], ZHANG Yan-Peng[1], DANG Jian-Wu[1], SHEN Li-Xin[2]
[1] School of Automation and Electrical Engineering
Lanzhou Jiaotong University
Lanzhou, 730070, China
[2]Department of Mathematics, Syracuse University
Syracuse, NY, 13244, USA
zhangzhenhai@mail.lzjtu.edu.cn

Journal of Digital
Information Management

***ABSTRACT**: Railway emergency management has gradually shifted from reaction planning to scenario planning when an emergency occurs. As such, an effective approach to integrated information management based on the "scenario–response" mode has become of considerable value in establishing a scenario model for railway emergency. First, the scenario elements of railway emergencies were classified by scenario analysis to set up a hierarchical network model for railway emergencies. After the analysis of the requirements, sources, and characteristics of information for scenario building, an approach to integrated information service for railway emergency scenarios based on the Service-Oriented Adaptive Agent Collaborative Model was proposed. Such approach could meet the requirements for autonomy and flexibility of an integrated information service under railway emergency conditions. Agent technology was adopted, and the Web service proxy was provided by the software agents in the proposed approach. The global scenario model was divided into many local procedure models in different departments, which distributed the control of integrated information service to numerous service nodes. Experimental results showed that the integrated information service system based on the proposed approach is significantly more autonomous and flexible than the integrated model based on the Business Process Execution Language for Web Services.*

Subject Categories and Descriptors
H.3.5 [Information Storage and Retrieval]: Online Information Services – Web-based services; **D.2.2 [Software Engineering]:** Design Tools and Techniques

General Terms: Railway, Information Management, Web services, Agent

Keywords: Railway Transportation, Emergency Information, Scenario Model, SOAACM, Integrated Service

Received: 18 August 2015, Revised 27 September 2015, Accepted 3 October 2015

1. Introduction

Railways allow for the conveyance of passengers and goods with the use of wheeled vehicles running on rails. Railways have become one of the most crucial transport systems in China and have played an indispensable role in the socioeconomic development of the country. Railway transport management has become increasingly challenging because of the growing demand for Chinese railway system that can accommodate high speed, high density and heavy load; such challenges in railway transport management increase the probability of the occurrence of emergency events [1]. In recent years, on the basis of previous experiences on accident disposal, many scholars have reported that the traditional "*planning–response*" mode is unable to meet the practical needs of emergency decision making, whereas scenario building can effectively reduce the incidence of such situations even with uncertain or incomplete information employed for decision making. Therefore, the research on emergency decision making based on the "scenario–response" mode has become the focus of considerable interest in the field of railway emergency management [2].

The scenario for emergency in decision making can be regarded as the prediction of the occurrence and development of events by determining the future development trend based on the analysis of the current state of events. During the process of scenario building for railway emergencies, massive multisource and heterogeneous information is required. Thus, integrated information service is one of the most crucial components of the scenario model for railway emergencies to ensure the effective application of the integrated logical data for visualization [3].

Currently, numerous mature frameworks and methods, such as middleware and data warehouse, are available in the field of integrated information. Although the integrated information approach based on the service-oriented architecture (SOA) is currently deemed a solution to the problem of multisource and heterogeneous information, it does not meet the requirements for autonomy and flexibility of railway emergency decision making [4]. Integrated information systems based on Web services have become one of the notable issues with respect to emerging information technologies, and integrated service based on Business Process Modeling has currently become the mainstream modeling ideal in integrated services, which include eFlow, Meteor-S, and Business Process Execution Language for Web Services (BPEL4WS). However, many deficiencies in design, operation, and evolution still remain in the existing models of integrated services. For example, the editing process of Extensible Markup Language is complex and the centralized control scheme has low efficiency; these deficiencies result in the inadaptability of integrated systems to various user needs and the dynamic environment [5]. A multi-agent technology can be considered an effective solution to the aforementioned problems. Therefore, the information services of the SOA will be taken as the central tasks with the coordination function of adaptive agents, and the integrated data set of a scenario can be established through the integrated information service to achieve unified storage, management, and services.

2. Network Model for Railway Emergency Scenarios

Railway emergency events result from various factors, and a chain of related events may occur. Thus, a railway emergency scenario should consist of influencing factors, event status, and event chains, and the different hierarchies of scenarios and the relationship between different scenarios can establish a hierarchical scenario network. The hierarchy of event scenarios is determined by the classification grain of each scenario and the extent of information acquisition. According to the hierarchical characteristics of emergency disposal for railway emergency events, a hierarchical network model for railway emergency scenarios, composed of an influencing factor layer, an event status layer, and an event chain layer, is proposed in the study. The proposed model is shown in Figure 1.

(1) Influencing Factor Layer
Numerous factors that induce, aggravate, and control events constitute the scenario network's influencing factor layer, which exists in the form of disaster factors. For example, common factors that contribute to rear-end collisions of trains include many scenarios, such as equipment failure, fault status without indication, dispatch order from the railway station supervisor, and operation of drivers in accordance with orders. The influencing factor layer may be assessed by conducting an analysis of the composition of scenarios.

(2) Event Status Layer
The event status layer shows the relationship between networks of different scenario statuses when an emergency occurs. This layer exists in the form of internal transition between event statuses. For instance, a scenario network is composed of various event statuses, such as the interference of lightning on the track circuit, station supervisor console without indication, and successful release of dispatch orders.

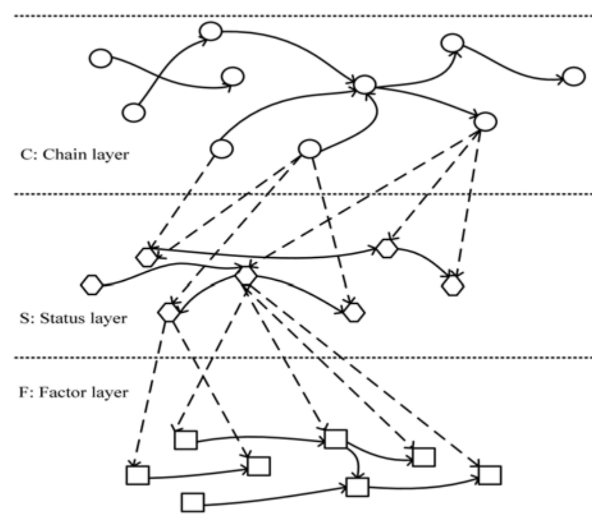

Figure 1. Hierarchical network model for railway emergency scenarios

(3) Event Chain Layer
The causality between events is shown in the event chain layer of the scenario networks, whose scenario transition is in the form of transition between events. For example, lightning events, track circuit failure events, and rear-end collisions of trains can form the scenario networks.

3. Information Sources for Scenario Building

Scenario building should be based on the abstraction and combination of information of event elements. The information types of a railway emergency scenario are numerous and may include planning information, on-the-spot information, resource information, and assistive decision-making information [6]. Event information can be classified into the following types:

(1) Time information: Time and time interval of the occurrence and development of events.

(2) Spatial information: Location, section, terrain, and distribution of key facilities when an event occurs.

(3) Facility information: Rrunning state and performance parameters of key facilities.

(4) Environmental Information: State of environmental elements, such as weather and traffic conditions when the event occurs.

(5) Information on agencies and staff: Organization staff, decision-making staff, and rescue staff responding to emergencies.

(6) Social Information: distribution of staff and resources to emergencies and the prevailing social situation.

When an emergency occurs, decision-making information about the emergency must be acquired in time. Such information comes from different business departments and is evidently heterogeneous. The use of such information may be optimized by integrating the required data from multiple heterogeneous sources and ensuring the consistency and integrity of data in the process of information integration. The interoperability of data can be easily achieved through unified and transparent interfaces for data access by users.decision-making staff, and rescue staff responding to emergencies.

4. Establishment of Service-Oriented Adaptive Agent Collaborative Model

An adaptive agent collaboration framework of integrated information services for railway emergency scenarios has been established to solve the problems of poor adaptability and low efficiency of integrated information of Web services and ultimately support the efficient, dynamic, and adaptive integration and collaboration among information services for decision making during railway emergencies [7]. The framework is based on multi-agent collaboration technology and is composed of software agents and Web services. In this study, a three-layer abstract architecture is adopted to define the Service-Oriented Adaptive Agent Collaborative Model (SOAACM). The relationships among the agents, Web services, and service procedures are depicted in Figure 2.

In Figure 2, the bottom layer is the Web service layer, which is composed of various Web services in railway computer networks with different functions for integrated service procedures. A Web service can be expressed as a tuple with four elements, as follows:

$$Webservice = \{Id,\ Description,\ Template,\ Qos\} \qquad (1)$$

where *Id* is the unique identifier of the Web service. *Description*, which is the description of the Web service, can be defined as follows

$$Description = \{fields,\ Input,\ Output,\ Function,\ Grain\} \qquad (2)$$

where the service fields, input parameters, output parameters, service function, and service grain are included. *Template* is the template of service. *Qos* is the quality of service defined as the vector of quality constraint, as follows:

$$Qos\_Cons = \{Pc, Tc, Rc, Ac, Ec\} \qquad (3)$$

where *Pc* is the price constraint, *Tc* is the time constraint, *Rc* is the reliability constraint, *Ac* is the availability constraint, and *Ec* is the credit constraint

The set of information services, *SV*, is the main encapsulation of the contents of integrated information and denotes the set of encapsulated information services for the integrated information about the entire scenario, which can form information services with different grains depending on the combination of information services. *SV* is defined as follows:

$$SV = \{SV_1 \times SV_1 \times \ldots \times SV_n,\ n \in Z\} \qquad (4)$$

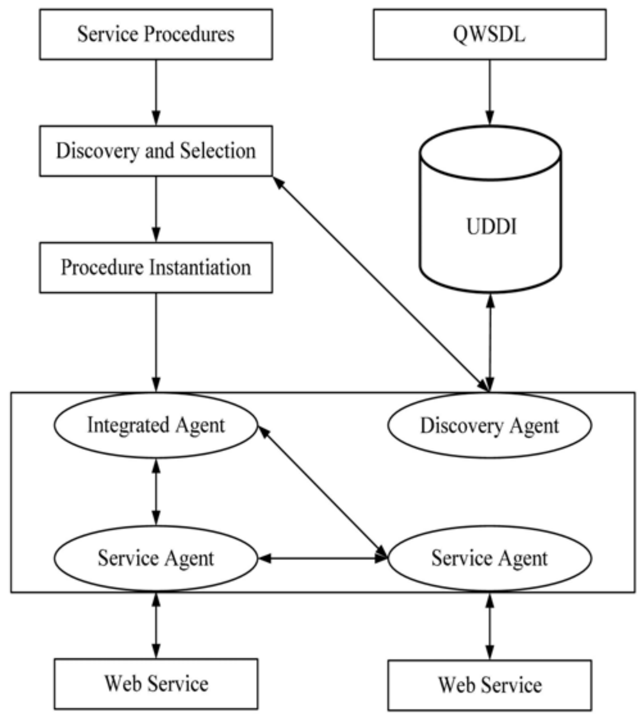

Figure 2. SOAACM

The execution of information service can be described as follows: If the premise condition *P* exists, then the integrated activity *A* can be executed by input parameters with type *I* to obtain output parameters with type *O* and execution result *R*, as follows:

$$SV = \exists P \rightarrow A\ (I(i_1, i_2, \cdots, i_m)) \rightarrow (O(O_1, O_2, \cdots, O_n) \cap R) \qquad (5)$$

The middle layer is the processing layer of agents. In this layer, the Web services are selected and customized in accordance with the functional requirements of the service procedures, which can be instantiated for the executable procedures of the integrated service to achieve message communication in the engine of integrated service, service call, and exceptional processing.

The top layer is the service procedure layer. In this layer, the service procedure of different functional components and the task order for emergency targets are formulated. The execution processes of *P* and *R* are also tracked.

(1) Service Agent
As a local executor of service procedures, the service agent is responsible for processing the Web service proxy, receiving the workflow script from the integrated agent, binding the specific Web service, implementing the related operation of Web service call, and interacting with other service agents. The service agent mainly consists of five function modules, namely, sub-procedure resolution, sub-procedure sending, agent session, service call, and adaptive service [8].

(2) Integrated Agent
The integrated agent is responsible for decomposing, sending, monitoring, and providing the result feedback of service procedures. The integrated service procedures from clients can be classified by using the description language of the integrated procedure BPEL4WS to generate the local service procedures of each service agent, which sends the script of the local service procedures to the corresponding service agent through the session with other service agents.

(3) Discovery Agent
The discovery agent is composed of service matching and service selection modules. The discovery agent accepts the abstract procedures from the scenario requirements, by which Web services can be searched and matched from the Universal Description, Discovery, and Integration registry. QWSDL is the acronym for Qos-based Web Service Description Language. According to the integrated information service algorithm based on QoS, the corresponding Web service can be selected to achieve binding of the Web service and instantiation of the abstract procedures [9].

5. Session between Agents

In the collaborative framework of SOAACM, agents can be classified into different types, including service agent, integrated agent, and discovery agent. Each type of agent has unique functions. Agents communicate through different sessions. Sessions have two main types, namely, the session between service and integrated agents and the session between service agents.

(1) Session Between Service And Integrated Agents
The session between service and integrated agents is conducted after the procedures of a Web service are decomposed to send the request to the service agent to execute the corresponding sub-procedures of the service. When the session starts, the service agent can opt to execute the session or not or to execute the session after a while. The service agent should respond to the execution request for service sub-procedures according to the current status of the service. A specific session is described in Figure 3.

(2) Session between Service Agents
When services have been provided, the service agent can determine the next service agent to implement the procedures of integrated service and send the notice of execution to the next service agent. The service agent can opt to execute the integrated service or not. If the service agent decides not to execute the integrated service, then the description of the next service agent is set to null and the integrated agent is informed to send a request to the discovery agent to search for a new service agent.

6. Experimental Simulations

In this study, the structure and performance of SOAACM are validated and compared with the BPEL4WS-based integrated model by conducting experimental simulations. Web services of integrated information for railway emergencies in three business departments have been established in the experiments for better performance comparison between the two schemes. The integrated information is composed of the information of electrical services monitoring, the information of line condition, and the information of train operation states. The numbers of sub-services in the Web services shown in Figure 4 are 100, 200, and 150. The SOAACM adopts the structure of the distributed model of integrated service, and message communication between agents adopts the direct transfer mode, thereby eliminating the need for indirect transfer through a certain central node. By contrast, the BPEL4WS-based integrated model employs the currently popular BPWS4J as execution engine, and messaging between Web services is executed by the central node BPWS4J. Thus, the service time is decided by the timing functions of the inner system and the system performance varies in terms of the number and size of the message sent per unit time.

(1) Effect of the increase in the number of concurrent requests on system performance

The experimental simulation is aimed at comparing the performances of the SOAACM-based distributed structure and BPEL4WS-based integrated model in terms of the capability to handle concurrent requests for service procedures. The size of the message between services is set to 1 KB. The variation curves of the integrated service time of both structures when the number of client requests gradually increases from 100 to 1,500 are shown in Figure 5, which reflect the load states of both systems. The curves in Figure 5 show that, when the number of requested procedures is low in the SOAACM-based distributed structure, the service time is added to the collaborative dispatching and procedure division between agents. Thus, the service time of the BPEL4WS-based integrated structure is shorter. However, with the increase in the number of requested procedures, bottleneck in

```
Session Protocol between Integrated Agent and Service Agent
Coordination_Agent_behavior(decomposed Flow fn)
    Am ← Get Service Agent(fn)
    response ← Request(Am)
    if response = accept then
    Send Flow(Am, fn)
    else if response = delay(t) then
    if Coordination_Agent can accept time delay then
    accept(Am)
    Send Flow(Am, fn)
    else
    reject(Am)
    search(substitute Am)
    end if
    else if response = reject then
    search(substitute Am)
    end if
Service_Agent_behavior(Service Request C)
    search Service Context
    if running_instance_number<allowed_instance_number then
    return accept
    else if request_instance_number>1 then
    return reject
    else
    return delay (next_instance_available_time)
    end if
```

Figure 3. Session between service and integrated agents

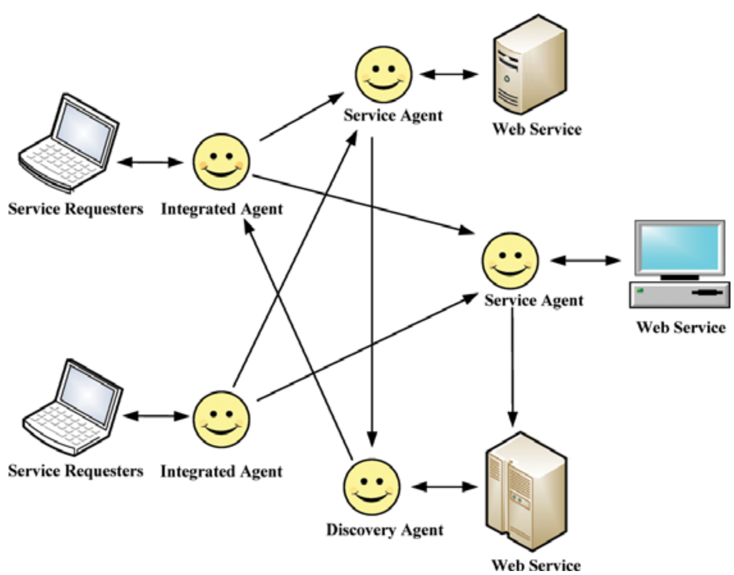

Figure 4. Experimental architecture of the system

dispatching emerges in the BPEL4WS-based integrated model, whose system load is evidently higher than that of the SOAACM-based distributed model. When the number of requested procedures increases to some extent, the load reaches the saturation state, and the load increase in each node in the SOAACM-based distributed structure is slower, as reflected in the gentle variation of its curve.

Emerging digital information technologies are giving ink-jet printing industry more opportunities to boost product quality. In this study, we focus on a digital information research for quantifying defect target tracking difficulty of the ink-jet print image sequence. This paper presents a defect-tracking algorithm under a particle-filtering framework for ink-jet printing videos, which can provide robust and accurate target localization and tracking. An observation model based on color histogram is built and adopted to calculate the likelihood of sample particles. A motion constrained resampling method to guide particles to appropriate states is then proposed. This method helps to solve the unpredictable abrupt motion efficiently and suppress the cluttered background properly. Experimental results on real ink-jet printing videos and comparisons with state-of-the-art algorithms demonstrate that the pro posed algorithm delivers more accurate tracking performance and is robust to abrupt motion, low SNR, as well as background changes.

(2) Effect of the size of the message sent between services on system performance

Another experimental simulation is conducted to compare the performance of the SOAACM-based distributed structure and the BPEL4WS-based integrated model in terms of the size of the message sent between services. The number of the reques0ted procedures is set to 200. Figure 6 shows the variation curves of the integrated service time of both structures when the size of the message sent between services gradually increases from 1 KB to 15 KB. The curves in Figure 6 indicate that, considering the direct data transfer mode used by the SOAACM-based distributed model, the data and the basic service consumption are transferred only once when data transfer occurs between adjacent service nodes. As a result, the integrated service time of the proposed model is reduced to almost half of the integrated service time of the BPEL4WS-based integrated model.

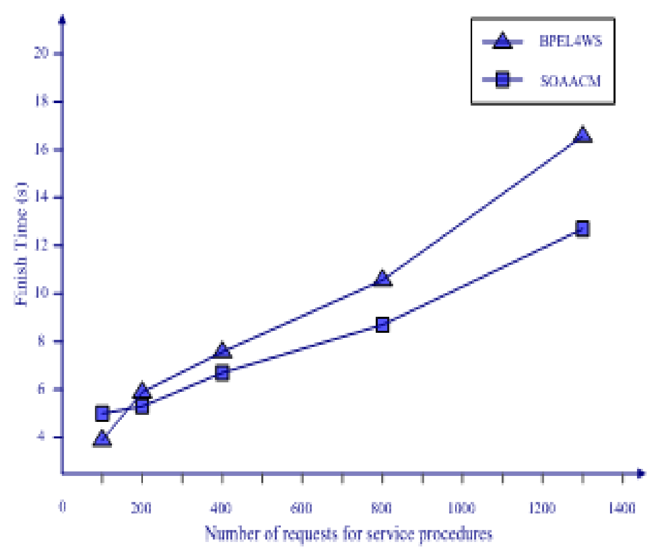

Figure 5. Performance analysis

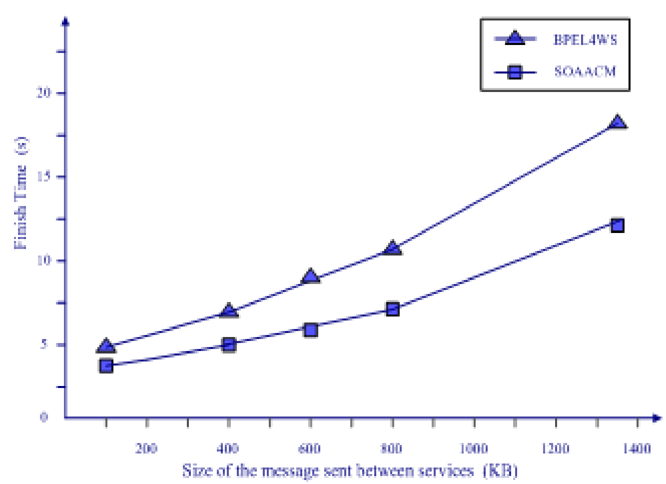

Figure 6. Performance analysis II

By comparing the graphs shown in Figures 5 and 6, we determined that the BPEL4WS-based integrated system is superior when the number of requested service procedures is low and the size of the message sent between services is small, whereas the SOAACM-based integrated service requires a longer time to finish procedure division. However, the SOAACM-based system is superior when the number of requested service procedures is high and the size of the message sent between services is large, reflecting the advantages of large throughput and short response time of the proposed system. Therefore, the BPEL4WS-based system is suitable for light-load situations, whereas the SOAACM-based system is suitable for the large-throughput and short-response-time situations. In railway emergency scenarios, massive integrated information is required in real time. Thus, the SOAACM-based distributed integrated service should be adopted to meet the requirements of various businesses. The distributed integrated service can distribute services to the relevant nodes and effectively mitigate the effect of integrated service on load under the condition of large concurrent requests to improve the performance of the entire system. Business data are also directly transferred between nodes without the help of a central node. As a result, network communication consumption is effectively reduced. Overall, the SOAACM-based system is more suitable for railway emergency scenarios that require an integrated service with massive amount of data.

6. Conclusion

On the basis of the requirements for the autonomy and flexibility of a large-scale integrated information service for railway emergency scenarios, the information sources of scenario building have been discussed and the current SOA-based integrated information approach has been analyzed in this study. A framework model of the integrated service based on SOAACM has also been proposed. In this approach, each role of the local procedure model is dynamically distributed to many service nodes for direct data transfer between service nodes. Through the software agent technology and the Web service proxy provided by agents, the SOAACM-based distributed integrated service system can be established. The experimental results and analysis show that the proposed approach is significantly more autonomous and flexible than BPEL4WS.

Acknowledgements

The study was supported by "*Science and TechnologyProject of China Railway Corporation, China (Grant No. 2014X008-F*)", "Natural Science Foundation for Young Scientists of Gansu Province, China (Grant No. 145RJYA242)" and "*Natural Science Foundation for Young Scientists of Lanzhou Jiaotong University, China (Grant No. 2013042*)".

References

[1] Lu, H. Z., Wang, F. Z., Wang, Y. J., Li, P. (2007). Research on the Framework of Railway Emergency Platform. C*hina Railways*, (6), 41-44.

[2] Qilin, S. H. U. (2012). Study on scenario evolvement and alternative generation of "*scenario-response*" decision-making in unconventional emergencies. *Journal of University of Science and Technology of China*, 42 (11), 936-941.

[3] Zheng, Y. J., Zhang, M. X., Ling, H. F., Chen, S. Y (2015). Emergency Railway Transportation Planning Using a Hyper-Heuristic Approach. *IEEE Transactions on Intelligent Transportation Systems,* 16 (1), 321-329.

[4] Adams, M., Ter Hofstede, A. H., Edmond, D., Van Der Aalst, W. M. (2006). Worklets: A service-oriented implementation of dynamic flexibility in workflows. In: On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE, Heidelberg, Berlin: Springer, 291-308.

[5] Curbera, F., Khalaf, R., Mukhi, N., Tai, S., Weerawarana, S (2003). The next step in Web services. *Communications of the ACM*, 46 (10), 29-34.

[6] Cheng, X. Q., Jia, L. M., Qin, Y., Zhang, Y. (2012). Research on railway emergency management. *Journal of the China Railway Society*, 34 (3), 7-13.

[7] Zhang, Z. H., Wang X. M., Zhang Y. P., Wang H. Y. (2013). Decision-Making Method of Railway Emergency Rescue Based on Hierarchical Agent. *Journal of Transportation Systems Engineering & Information Technology*, 13 (2) 141-146.

[8] Shi, Y., Huang, G. A., Ye, W., Zhang, L., Shi, B. (2006). Automatic composition of Web services based on task dependency specification. *Journal of Computer Research and Development*, 43 (12), 2110-2116.

[9] Yu, T., Zhang, Y., Lin, K. J. (2007). Efficient algorithms for Web services selection with end-to-end QoS constraints. *ACM Transactions on Web (TWEB)*, 1 (1), 6-26.